Improved Environmental Adaption Method with Real Parameter Encoding for Solving Optimization Problems

Tribhuvan Singh, Ankita Shukla and K.K. Mishra

Abstract Environmental Adaption Method (EAM) was proposed by K.K. Mishra et al. in 2011. Further an improved version of EAM with binary encoding was proposed in 2012, known as Improved Environmental Adaption Method (IEAM) with some changes in adaption operator. IEAM uses adaption, alteration, and selection operators to generate new population. In this paper, we have implemented a real parameter version of IEAM. In IEAM, adaption window of variable bandwidth was used for evolution of solutions, due to this particles could not evolve properly in entire search space. Here, we have used adaption window of fixed bandwidth for proper evolution of solutions. Performance of Improved Environmental Adaption Method with real parameter encoding (IEAM-RP) is compared with other nature-inspired optimization algorithms on Black Box Optimization Test-bed at dimensions 2D, 3D, 5D, and 10D on a set of 24 benchmark functions. It is found that IEAM-RP performs better than other state-of-the-art algorithms.

Keywords Randomized algorithm · EAM · IEAM · Adaption factor · Mobility factor

1 Introduction

To solve optimization problems, randomized algorithms are always a good choice. Randomized algorithms are useful when direction of search is not known in the beginning. It starts search with random search space and finds optimal solution in

T. Singh (✉) · A. Shukla · K.K. Mishra
Computer Science and Engineering Department, MNNIT Allahabad,
Allahabad 211004, India
e-mail: tribhuvan.mnnit@gmail.com

A. Shukla
e-mail: shuklankita321@gmail.com

K.K. Mishra
e-mail: kkm@mnnit.ac.in

S.K. Bhatia et al. (eds.), *Advances in Computer and Computational Sciences*,
Advances in Intelligent Systems and Computing 554,
https://doi.org/10.1007/978-981-10-3773-3_2

minimum time. There are various randomized algorithms that use nature inspired search technique such as PSO [1], EA, GA. Environmental Adaption Method (EAM), a randomized optimization algorithm is derived from natural phenomenon [2]. EAM is based on adaptive learning theory. As environment changes either a species can extinct or evolve. Key to survival is to adapt to environmental changes as earlier as possible. EAM has three operator namely adaption, alteration, and selection. After applying these three operator on initial population, next generation population will be generated. In EAM, diversity provided by alteration operator was not enough to solve multimodal problems. To improve diversity in solutions, a new version of EAM named as IEAM was proposed in 2012. New version of any algorithm can be created either by changing its operator or by parameter tuning. In IEAM, there are few changes in operators as well as there is fine-tuning of parameters. In IEAM, adaption operator has best particle which will explore other good regions in the search of optimal solution and other than best particles will use adaption window of variable bandwidth. Due to this, sometimes, solutions are not able to evolve properly. To overcome this problem, a new version of IEAM with real parameter encoding is proposed in this paper. In this paper, we mapped formulae of IEAM into real parameter version. Unlike IEAM, in IEAM-RP, an adaption window of fixed bandwidth is used, due to which solutions are able to properly evolve in the entire search space.

Remaining of this paper is organized as follows. Section 2 provides background details. Section 3 presents our proposed algorithm. Section 4 covers experimental setup. In Sect. 5, result analysis is done and in the last section conclusions are drawn.

2 Background Details

2.1 EAM

EAM uses the theory of adaptive learning [2]. EAM is a population-based algorithm. It uses binary encoding. It uses three operators namely adaption, alteration, and selection. It starts with randomly initialized population. All species will try to improve its phenotypic structure so as to adjust with new environmental conditions with adaption operator and those species which are not able to survive in new environment will be destructed. After this, alteration operator is applied to provide diversity to the solutions. When both of these operators are applied on old population, an intermediate population is generated. After that selection operator is applied. It combines old population and intermediate population and selects best individuals equal to initial population size on the basis of fitness. This process is repeated until we get the best solution or maximum number of iterations has been reached. There were following shortcomings of EAM.

- EAM works with binary encoding. So each time there is need of decimal to binary conversion, which is an extra overhead.
- In higher dimensions, convergent rate is not good enough and prone to Stagnation.

2.2 IEAM

To improve convergence rate of EAM and preventing it to converge on local optimal solution, Improved Environmental Adaption Method (IEAM) was proposed in 2012 by Mishra et al [3, 5]. IEAM uses basic framework of EAM except there is a change in adaption operator. In EAM, each solution updates its structure on the basis of environmental fitness and current fitness. There is no concept of best particles. Unlike EAM, IEAM uses concept used by PSO to update its phenotypic structure. As in PSO, particle updates its structure on the basis of its personal best fitness and global best fitness [1]. In the same way, in IEAM, direction of search is guided by best particle having optimal fitness so far and particle's own fitness. To promote diversity, best particle will explore whole search space and remaining particles will be guided by the best particle. Here, adaption operator is used for exploitation as well as exploration of search space. In IEAM, there is fine-tuning of parameters. With proper setting of parameters global optimal solution can be achieved in early generations.

3 Proposed Approach

Although IEAM has very high convergence rate but it is binary coded. The outcome of binary coding based optimization algorithm depends on how many solutions are considered. If very less number of solutions are taken then there may be a huge difference in obtained solution and desired solution. There is a need of binary to decimal conversion each time which is an extra overhead. There is a need of large number of bits to get accurate results in higher dimensions. To solve these problems, a real parameter version of IEAM, IEAM-RP is suggested. IEAM-RP uses basic framework of IEAM. Like PSO, IEAM uses concept of best particle. In IEAM, Best particle uses the following formula to generate new position

$$P_{i+1} = [\alpha \ * \ (P_i)^{F(X_i)/F_{avg}} + \beta]\%2^l, \tag{1}$$

where P_i is the position value of a particle that is updating its structure. α and β are tuning parameters and l is number of bits. $F(X_i)$ is fitness of *ith* particle and F_{avg} is current environmental fitness. Particles other than best update their positions with formula given below

$$P_{i+1} = [\alpha \ * \ (P_i)^{F(X_i)/F_{avg}} + \beta * [(G_b - P_i) + (P_b - P_i)]]\%2^l \tag{2}$$

where G_b is the position vector of best particle and P_b is personal best position vector of the particle that wants to change its position. Values of α and β are taken between 0 to 1.

In real coded version of IEAM, there is no need of binary to decimal conversion. Since we are dealing with real parameters directly, we do not need l (number of bits) any more. Like IEAM, in IEAM-RP clamping is done if the solutions move beyond the search space, but here we do not need modulus operator. Unlike IEAM, in IEAM-RP, we multiply old position by $F(X_i)/F_{avg}$ rather than putting it into the exponent, because if we put this term in exponent then it may result in a very large number or sometimes it may generate a complex number. So, in IEAM-RP best particle will use the following adaption operator

$$P_{i+1} = P_i * F(X_i)/F_{avg} + \beta, \tag{3}$$

where β is any random number between 0 to 1. Like IEAM, other than best particle will move in the direction of best particle to attain phenotypic structure of best particle. In binary IEAM, adaption window was different for each solution, as shown in Eq. 2. Due to variable bandwidth of adaption window, some solutions could not exploit the region properly. To resolve this issue, IEAM-RP uses adaption window of fixed bandwidth (difference between Best_Position and Worst_Position). With fixed bandwidth now solutions exploit properly in the region. Here, formula for adaption of other than best particles is mapped in the following manner

$$P_{i+1}P_i + \beta * (Best\_Position - Worst\_Position) \tag{4}$$

In proposed algorithm, alteration operator of IEAM is not used, because adaption operator here is powerful enough to produce diverse solutions. But alteration operator is not excluded from basic IEAM. In future, if any application needs more diversity than it is provided by IEAM-RP, alteration operator can be added to IEAM-RP. IEAM-RP uses selection operator in the same way as it was used in IEAM. Selection operator is used to select the best solutions equal to the number of initial population size from parent population and offspring. In this way, IEAM-RP ensures elitism. This process continues until stopping criteria is met.

3.1 Algorithms

Adaption Operator: Notations

P_i	Current Population
$Fitness$	Fitness of particle
P_{i+1}	Adapted Population
β	Randam Number between 0 to 1
F_{avg}	Environmental Fitness
AF	Adaption Factor
MF	Mobility Factor

Algorithm 1 Adaption(P_i,Fitness)

1: MF = best_position-worst_position
2: **for** each individual in P_i **do**
3: $AF = Fitness/F_{avg}$
4: **end for**
5: **for** each individual in P_i **do**
6: $P_{i+1} = AF * P_i + \beta$ ▷ Best particle will adapt with this formula
7: $P_{i+1} = P_i + MF * \beta$ ▷ Other particles will adapt with this formula
8: **end for**
9: Clamp the position of particles if they move beyond the range
10: return P_{i+1}

Selection Operator:Notations

T_POP	Temporary Population
S_POP	Population after Sorting
P_{i+1}	Adapted Population
P_i	Current Population
ps	Population Size

Algorithm 2 Selection(P_i,P_{i+1},ps)

1: T_POP = merge(P_i, P_{i+1})
2: S_POP = sort(T_POP)
3: P_i = select ps fittest individual from S_POP
4: return P_i

IEAM-RP: Notations

POP_i	Population at i^{th} generation
P_{i+1}	Adapted Population
MaxGen	Maximum number of generations
$IPOP_i$	Intermediate population at i^{th} generation

Algorithm 3 IEAM-RP

1: Initialize Population POP_1 randomly
2: **repeat**
3: **for** i = 1 to MaxGen **do**
4: Evaluate Fitness of each particle
5: P_{i+1} = Adaption($POP_i, Fitness_i$)
6: POP_{i+1} = Selection(POP_i, P_{i+1})
7: **end for**
8: **until** stopping criteria is not met or optimal solution is not found

3.2 Details of Algorithm

IEAM-RP uses the following steps to generate optimal solution

1. **Initialization Phase**: All solutions are randomly initialized in search space. This creates population for the first generation. Variables are also initialized in this phase
ps = 100 * DIM;
fbest = inf;
xbound = 5;
fun = FUN;
Dim = DIM;
oldpop = 2 * xbound * rand(ps,DIM) −xbound*ones(ps,Dim);
fitness = feval(fun,oldpop');
fitness = fitness';
maxfunevals = min(1e5 * DIM, maxfunevals);
maxiterations = maxfunevals;

2. **Next Generation Creation**: IEAM-RP uses two basic operators of IEAM that are adaption and selection. Functionality of each operator is explained below

   (a) **Adaption**: Adaption operator both explore and exploit the problem search space. Here, we use the term adaption factor (AF) to represent $F(X_i)/F_{avg}$ which is used for adaption of best particle, and difference of Best_Position and Worst_Position is called as Mobility Factor (MF). MF is responsible for adaption of other than best particles. Best Particle will move in whole search space with AF. AF depends on environmental fitness i.e., average fitness of all particles and their own fitness. Other than best particles will try to attain phenotypic structure of best particle with the help of mobility factor. Here, MF is responsible for providing fixed bandwidth. Function for adaption operator is given below
   $function[new\_pop, fitad] = adaption(oldpop, xbound, Dim, fitness, ps)$
   $xmin = -xbound * ones(1, Dim);$
   xmax = xbound * ones(1,Dim);
   fitad = fitness; favg = mean(fitad);
   favg = favg*ones(ps,Dim);
   fitad3 = repmat(fitad,1,Dim); c = fitad3./favg;
   mb = oldpop(1,:)−oldpop(ps,:);
   $new\_pop(1, :) = c(1, :). * oldpop(1, :) + rand(1, Dim);$
   for h = 2:ps
   $new_pop(h, :) = oldpop(h, :) + mb. * rand(1, Dim);$end
   $s = new\_pop < repmat(xmin, ps, 1);$
   $new\_pop = (1 - s). * new_pop + s. * repmat(xmin, ps, 1);$
   $b = new\_pop > repmat(xmax, ps, 1);$
   $new_pop = (1 - b). * new_pop + b. * repmat(xmax, ps, 1);$end
   (b) **Selection**: This works in the same way as it was in binary encoded IEAM. In each generation, intermediate population and old population are merged, then best individuals equal to initial population size are chosen on the basis

of fitness to generate new population. Function for selection operator is given below

$function[sel\_pop, fitsel] = selection(oldpop, new1\_pop, ps, fun, Dim, fitad1)$
fin = cat(2,oldpop,fitad1);
$x = feval(fun, new1\_pop');$
$fin1 = cat(2, new1\_pop, x');$
marge = cat(1,fin,fin1);
$final\_sort = sortrows(marge, Dim + 1);$
$sel\_pop = final\_sort(1 : ps, 1 : Dim + 1);$
$fitsel = sel\_pop(:, Dim + 1);$
$sel\_pop = sel\_pop(1 : ps, 1 : Dim);$

3. **Generation step and evolution**: In each generation, adaption and selection is applied on parent population to generate offspring.This process is repeated until either maximum number of generations are reached or we get desired optimal solution.

4 Experimental Setup

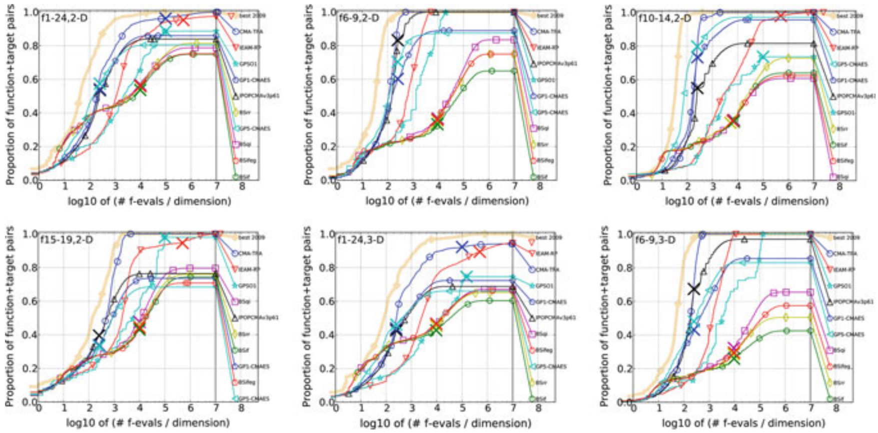
For experiments, Black Box Optimization Test-bed is used where search domain for all 24 benchmark functions is [−5, 5]. The algorithm is tested for dimensions 2-D, 3-D, 5-D, and 10-D with population size 100 * DIM without any restart mechanism.

5 Result Analysis

The performance of IEAM-RP is compared with other optimization algorithms like CMA-TPA, GPSO1, GP1-CMAES, IPOPCMAv3p61, BSrr, GP5-CMAES, BSqi, BSifeg, and BSif. The rank of IEAM-RP for separ, lcond, hcond, multi, multi2, and for all and for dimension 2, 3, 5 and 10 is given in Table 1. To check the efficiency of IEAM-RP, proposed algorithm has been applied to standard 24 COCO benchmark functions [4, 6, 7]. From Table 1 it is obvious that adaption operator of proposed algorithm provides solutions that are divesed enough and due to this it gives better results as compared to other algorithms.

Table 1 IEAM-RP rank in different dimensions

	separ	lcond	hcond	multi	multi2	all
2*D*	5	2	2	2	2	2
3*D*	5	2	2	2	1	1
5*D*	5	2	3	2	2	2
10*D*	5	2	5	2	2	2



6 Conclusion

A real coded version of IEAM is proposed here. It is different from IEAM in two ways. First it works with real parameters. Second, in adaption operator, we have used adaption window of fixed bandwidth for evolution of other than best particle which makes adaption operator more powerful. Results show that IEAM-RP outperform other state-of-the art algorithms.

References

1. J. Kennedy, R. Eberhart, "Particle swarm optimization," Neural Networks, 1995, IEEE International Conference on Neural Networks, vol. 4, no., pp.1942–1948 vol.4, Nov/Dec 1995.
2. Mishra, K. K., Shailesh Tiwari, and A. K. Misra. "A bio inspired algorithm for solving optimization problems." In Computer and Communication Technology (ICCCT), 2011 2nd International Conference on, pp. 653–659. IEEE, 2011.
3. Mishra, K. K., Shailesh Tiwari, and A. K. Misra. "Improved Environmental Adaption Method for Solving Optimization Problems." In Computational Intelligence and Intelligent Systems, pp. 300–313. Springer Berlin Heidelberg, 2012.
4. http://coco.gforge.inria.fr/.
5. Mishra, K.K., Shailesh Tiwari and A.K. Misra. "Improved environmental adaption method and its application in test case generation" In Journal of Intelligent and Fuzzy Systems xx (20xx) x–xx DOI:10.3233/IFS-141195 IOS Press.
6. N. Hansen et al. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009. Updated February 2010.
7. N. Hansen et al. Real-parameter black-box optimization benchmarking 2012: Experimental setup. Technical report, INRIA, 2012.

Springer

http://www.springer.com/978-981-10-3772-6

Advances in Computer and Computational Sciences
Proceedings of ICCCCS 2016, Volume 2
Bhatia, S.K.; Mishra, K.K.; Tiwari, S.; Singh, V.K. (Eds.)
2018, XXIV, 713 p. 254 illus., Softcover
ISBN: 978-981-10-3772-6